Revealing the Critical Role of Human Performance in Software

**IT'S TIME TO REVISE
OUR APPRECIATION
OF THE HUMAN SIDE
OF INTERNET-FACING
SOFTWARE SYSTEMS.**

DAVID D. WOODS AND JOHN ALLSPAW

People are the unique source of adaptive capacity essential to incident response in modern Internet-facing software systems. The collection of articles in this issue of *acm queue* seeks to explore the forms of human performance that make modern business-critical systems robust and resilient despite their scale and complexity.

In the first of four articles in this issue, Richard Cook reframes how these Internet-facing systems work through his insightful “Above the Line/Below the Line” framing that connects human performance to software tooling. He connects human performance above the line to technology performance below the line of representation.

Then Marisa Grayson considers a key function above the line by studying the cognitive work of anomaly response, particularly how hypotheses are explored during incident response.

In her article, Laura Maguire expands the above-the-line frame by examining what coordination looks like across multiple roles when events threaten service outages, especially how people adapt to control the costs of this coordination.

Finally, J. Paul Reed broadens the perspective to reveal

Without the cognitive work that people engage in with each other, all software systems eventually fail (some catastrophically!).

the factors that affect how learning from incidents can be narrow and reactive or broad and proactive. Broad, proactive learning keeps pace with change, continually recharging the sources of adaptive capacity that lead to resilient performance.

The articles all highlight the idea that incidents are opportunities to update and revise models of the ways organizations generate and sustain adaptive capacities to handle surprising challenges as systems grow and operate at new scales.

While it's reasonable for software engineering and operations communities to focus on the intricacies of technology, there's not much attention given to the intricacies of how people do their work. This is a critical gap since modern business-critical systems work as well as they do because of the adaptive capabilities of *people*. To be more specific: Without the *cognitive work* that people engage in with each other, all software systems eventually fail (some catastrophically!).¹

Business-critical software systems necessarily increase in complexity as they become more successful. This complexity makes these systems inherently messy, so that surprising incidents are part and parcel of the capability to provide services at larger scales and speeds (see “Resilience is a Verb” for a basic introduction¹⁰). Incidents will continue to present challenges that require resilient performance, regardless of past reliability statistics.

Studies in resilience engineering^{2,9} reveal that people produce resilient performance by (1) doing the cognitive work of anomaly response; (2) coordinating joint activity during events that threaten service outages; and (3)

revising their models of how the system actually works and malfunctions using lessons learned from incidents. People's resilient performance compensates for the messiness of systems, despite constant change.

If you take the view that systems are up, working, and successful because of the adaptive capacity that people have, then incidents can be reframed as ongoing opportunities to update and revise mental models as the organization/technology/infrastructure changes, grows, and scales.⁴

HUMAN PERFORMANCE AND SOFTWARE ENGINEERING CDI (critical digital infrastructure) encompasses the facilities that engineers use to develop, change, update, and operate software that enables valuable services. This includes all the components needed to create the value that businesses provide to customers: the technology stack, code repositories, data sources, and a host of tools for testing, monitoring, deployment, and performance measurement, as well as the various ways of delivering these services.

In his opening article, Cook points out that discussions focused solely on the technology miss what is actually going on in the operations of Internet-facing applications. Figure 1 in Cook's article reveals the *cognitive work* and *joint activity* that go on above the line and places the technology and tooling for development and operations below the line. The "line" here is the *line of representation*. No one can directly inspect or influence the processes running below the line; all understanding and action are mediated through *representations*.

The above-the-line area in the diagram includes the people who are engaged in keeping the system running and extending its functionality. They are the ones preparing to deploy new code, monitoring system activities, re-architecting the system, etc. These people ask questions such as: What's it doing now? Why is it doing this? What's it going to do next? This cognitive work—observing, inferring, anticipating, planning, intervening, etc.—is done by interacting, not with the things themselves, but with *representations* of them. Interestingly, some representations (e.g., dashboards) are designed by (and for) software engineers and other stakeholders.

Notice that all the above-the-line actors have mental models of what is *below* the line. These vary depending on role and experience, as well as on their individual perspectives and knowledge. Notice that the actors' mental models are different. This is because there are general limits on the fidelity of models of complex, highly interconnected systems.⁸ This is true of modern software systems and is demonstrated by studies of incident response; a common statement heard during incidents or in the postmortem meetings afterward is, “I didn't know it worked that way.”⁹

SYSTEMS ARE MESSY

Systems are developed and operate with finite resources, and they function in a constantly changing environment. Plans, procedures, automation, and roles are inherently limited; they cannot encompass all the activities, events, and demands these systems encounter. Systems operate under multiple pressures and virtually

It is people who adapt to meet the inevitable challenges, pressures, tradeoffs, resource scarcity, and surprises that occur.

always in degraded mode.¹⁰

The adaptive capacity of complex systems resides in people. It is *people* who adapt to meet the inevitable challenges, pressures, tradeoffs, resource scarcity, and surprises that occur. A slang term from World War II captures both the state of the system and the acceptance of the people who made things work: SNAFU (situation normal, all fouled up). With this term, soldiers were acknowledging that this is the usual status and that their jobs were to make the flawed and balky parts work. If SNAFU is normal, then SNAFU catching is essential—resilient performance depends on the ability to adapt outside of standard plans, which inevitably break down.

However technologically facilitated, *SNAFU catching* is a fundamentally human capability that is essential for viability in a world of change and surprise. Some people in some roles provide the essential adaptive capacity for SNAFU catching, though the catching itself may be local, invisible to distant perspectives, or even conducted out of organizational view.⁶

Surprises in complex systems are inevitable. Resilience engineering enhances the *adaptive capacity* needed for response to surprises. A system with adaptive capacity is poised to adapt. It has some readiness to change how it currently works—its models, plans, processes, behaviors—when it confronts anomalies and surprises.⁸ Adaptation is the potential to modify plans to continue to fit changing situations. NASA's Mission Control Center in Houston is a positive case study for this capability, especially how Space Shuttle mission controllers developed skill at handling anomalies, expecting that the next anomaly they

would experience was unlikely to match any of the ones from the past that they had practiced or experienced.⁷

IT-based companies exist in a pressurized world where technology, competitors, and stakeholders change. Their success requires scaling and transforming infrastructure to accommodate increasing demand and build new products. These factors add complexity (e.g., having to cope with incident response involving third-party software dependencies) and produce surprising anomalies.^{1,9} Knowing they will experience anomalies, IT-based companies, organizations, and governments need to be fluent at change and poised to adapt.

ANOMALY RESPONSE

The articles by Marisa Grayson and Laura Maguire reveal that IT-incident response is an example of the cognitive work of anomaly response.¹¹

Grayson focuses on the general function of *hypothesis exploration* during anomaly response. Hypothesis exploration begins with recognition of an anomaly (i.e., a difference between what is observed and the observer's expectations). Those expectations are derived from the observer's model of the system and the specific context of operations. Anomaly recognition in large, interconnected, and partially autonomous systems is particularly difficult. Sensemaking is challenging when monitoring a continuous flow of changing data about events that might be relevant. For many Internet-facing business systems this is the norm: Data streams are wide and fast-flowing; normal variability is high; alert overload is common; operations and observations, as well as technology, are highly

The quality of anomaly response, however, is directly related to the ability to generate and consider a wide range of hypotheses and to revise hypotheses as the situation changes over time.

distributed. To make matters worse, the representations typically available require long chains of inference rather than supporting direct visualization of anomalous behaviors.

Grayson's results show how practitioners generate, revise, and test potential explanations that could account for the unexpected findings. She developed a method to diagram and visualize hypothesis exploration based on the above-the-line/below-the-line framework.

Her charts reveal the typical flow of exploration where multiple hypotheses are generated to account for the anomalies, and the hypotheses in this set change over time. As response teams converge on an assessment of the situation, they frequently revise what are deemed candidate hypotheses and their relative confidence across the possibilities. In her study, Grayson found that sometimes a hypothesis that was considered to be confirmed was overturned as new evidence later came to the fore.

In hindsight, people focus on the answer that resolved the incident. The quality of anomaly response, however, is directly related to the ability to generate and consider a wide range of hypotheses and to revise hypotheses as the situation changes over time—for example, when interventions to resolve problems end up producing additional unexpected behavior.

CONTROLLING THE COSTS OF COORDINATION IN JOINT ACTIVITY

Maguire expands what goes on above the line by examining how people adapt so they can control the cost of

coordination for joint activity.

The value of coordination across roles and perspectives is well established. Handling anomalies in risky worlds such as space mission operation centers is one example.⁷ But studies of joint activity also reveal that the costs of coordination can offset the benefits of involving multiple people and automation in situation management.⁵ This earlier research looked at anomaly response anchored in physical control rooms where responders were collocated in open workspaces.

Internet-facing software systems are managed differently, as the norm is for responders to be physically distributed. People connect via ChatOps channels, unable to observe each other. The cognitive costs of coordination are greater for geographically distributed groups. Maguire's article describes how this both enables and constrains joint activity. For example, growth has led to third-party software dependencies that require coordination across organization (and company) boundaries during anomaly response.

In her research, Maguire asks the question: What do practitioners do to control the costs of coordination as they carry out anomaly response under uncertainty, risk, and pressure? Her results are based on studying how software engineers experience these "costs" across a set of incident response cases. They highlight the shortcomings of traditional ways of coordinating roles and managing the costs of coordination (e.g., incident commander, disciplined procedure-following (based on an incident command system), and efforts to use IT prosthetics such as bots. Maguire's work reveals how

people adapt when the costs of coordination become larger. Understanding these adaptations can help in the design of effective tools, alter roles, and build organizational frameworks that enhance joint activity and reduce the costs of coordination during incident response.

LEARNING WHAT MAKES INCIDENT RESPONSE WORK

There is a significant gap between how we imagine incidents occur (and are resolved) and how they *actually* occur.³ The final article by J. Paul Reed considers how organizations learn to close this gap. Reed's research highlights an important but often invisible driver of work above the line—the ways people capture lasting memories of past incidents and how these are used by those not present or involved with handling them at the time. How do people come to understand what happened? How do they share attributions about why it happened? Why do some incidents attract more organizational attention than others?

Organizations usually reserve limited resources to study events that have resulted in (or come close to) significant service degradation. Social, organizational, and regulatory factors constrain what learning is possible from such events. In contrast, *proactive* learning about resilient performance and adaptive capacities focuses on how cognitive work usually goes well despite all of the difficulties, limited resources, tradeoffs, and surprises. The data and analyses in previous reports illustrate the potential insights to be gained from in-depth examination of the cognitive work of incident response.^{2,9}

CONCLUSION

Together, the four articles in this issue provide a sketch of what is happening above the line of representation, especially during incident responses. These activities are essential to building, fielding, and revising the modern information technology on which our society increasingly depends. Understanding how people detect anomalies, work together resolving incidents, and learn from those experiences is essential for having more resilient systems in the future.

The intimate relationship between human expertise and the technological components of modern systems defies linear decomposition. As Cook shows, there is really only one system here—how the system works and evolves depends on an awareness of how people’s capacity to adapt is sometimes facilitated and at other times frustrated by the technology. The articles by Grayson, Maguire, and Reed demonstrate how looking at incidents through the lens of cognitive work, joint activity, and adaptive capacities provides new insights about how this human-technology system really works. Incidents are challenges that reveal the system doesn’t work the way it has been imagined. The experience of the incident and post-incident inquiry offer learning opportunities highlighting where mental models need revision.

The articles go further, though. Together they highlight how everyone’s mental models about Internet-facing software systems are in need of significant revision. Human cognitive, collaborative, and adaptive performance is central to software engineering and operations. As the scale and complexity of the software systems necessary

to provide critical services continue to increase, what goes on above the line will remain central to all stories of growth, success, precariousness, and breakdown.

Understanding, supporting, and sustaining the capabilities above the line require all stakeholders to be able to continuously update and revise their models of how the system is messy and yet usually manages to work. This kind of openness to continually reexamine how the system really works requires expanding the efforts to learn from incidents. These articles provide tangible paths all can follow to learn how to learn from incidents.

References

1. Allspaw, J. 2016. Human factors and ergonomics practice in web engineering and operations: navigating a critical yet opaque sea of automation. In *Human Factors and Ergonomics in Practice*, ed. S. Shorrock and C. Williams, 313-322. Boca Raton, FL: CRC Press (Taylor & Francis).
2. Allspaw, J. 2015. Tradeoffs under pressure: heuristics and observations of teams resolving Internet service outages. Master's thesis. Lund, Sweden: Lund University.
3. Allspaw, J. 2018. Incidents as we imagine them versus how they actually are. PagerDuty Summit 2018. YouTube; <https://www.youtube.com/watch?v=8DtzmV1jyQ>.
4. Allspaw, J., Cook, R. I. 2018. SRE cognitive work. In *Seeking SRE: Conversations About Running Production Systems at Scale*, ed. D. Blank-Edelman, 441-465. O'Reilly Media.
5. Klein, G., Feltovich, P. J., Bradshaw, J. M., Woods, D. D.

2005. Common ground and coordination in joint activity. In *Organizational Simulation*, ed. W. Rouse and K. Boff, 139-184. Wiley.
6. Perry, S. J., Wears, R. L. 2012. Underground adaptations: cases from health care. *Cognition, Technology & Work* 14(3), 253–60; doi.org/10.1007/s10111-011-0207-2.
7. Watts-Perotti, J., Woods, D. D. 2009. Cooperative advocacy: a strategy for integrating diverse perspectives in anomaly response. *Computer Supported Cooperative Work: The Journal of Collaborative Computing* 18(2), 175–98.
8. Woods, D. D. 2015. Four concepts of resilience and the implications for resilience engineering. *Reliability Engineering and Systems Safety* 141, 5-9; doi:10.1016/j.res.2015.03.018.
9. Woods D. D. 2017. Stella Report from the SNAFUcatchers Workshop on Coping with Complexity; <https://snafucatchers.github.io/>.
10. Woods, D. D. 2018. Resilience is a verb. In *IRGC Resource Guide on Resilience (vol. 2): Domains of Resilience for Complex Interconnected Systems*, ed. B. D. Trump, M.-V. Florin, and I. Linkov. Lausanne, Switzerland: EPFL International Risk Governance Center. https://www.researchgate.net/publication/329035477_Resilience_is_a_Verb.
11. Woods, D.D., Hollnagel, E. 2006. *Joint Cognitive Systems: Patterns in Cognitive Systems Engineering*. Boca Raton, FL: CRC Press [Taylor & Francis].

John Allspaw has worked in software systems engineering and operations for over twenty years. John's publications

include the books The Art of Capacity Planning (2009) and Web Operations (2010) as well as the foreword to The DevOps Handbook. His 2009 Velocity talk with Paul Hammond, "10+ Deploys Per Day: Dev and Ops Cooperation" helped start the DevOps movement. John served as CTO at Etsy and holds an MSc in human factors and systems safety from Lund University.

David D. Woods, *Professor, Integrated Systems Engineering, the Ohio State University, has studied human coordination with automated and intelligent systems in almost every high-risk complex setting over the last 40 years. He developed resilience engineering on the dangers of brittle systems and the need to invest in sustaining sources of resilience beginning in 2000-2003 as part of the response to several NASA accidents. His books include Behind Human Error (1994/2010), Resilience Engineering: Concepts and Precepts (2006), and Joint Cognitive Systems (Foundations, 2005 / Patterns, 2006). He is Past President of the Human Factors and Ergonomics Society and of the Resilience Engineering Association.*

Copyright © 2019 held by owner/author. Publication rights licensed to ACM.